CS 188: Artificial Intelligence
Spring 2011

Lecture 20: Naïve Bayes
4/11/2011

Pieter Abbeel – UC Berkeley
Slides adapted from Dan Klein.

Announcements

- W4 due right now
- P4 out, due Friday
- First contest competition

Today

- Naïve Bayes
  - Inference
  - Parameter estimation
  - Generalization and overfitting
  - Smoothing
- General classification concepts
  - Confidences
  - Precision-Recall

Example Classification Tasks

- In classification, we predict labels y (classes) for inputs x
- Examples:
  - Spam detection (input: document, classes: spam / ham)
  - OCR (input: images, classes: characters)
  - Medical diagnosis (input: symptoms, classes: diseases)
  - Automatic essay grader (input: document, classes: grades)
  - Fraud detection (input: account activity, classes: fraud / no fraud)
  - Customer service email routing
  - ... many more
- Classification is an important commercial technology!

Bayes Nets for Classification

- One method of classification:
  - Use a probabilistic model!
  - Features are observed random variables F
  - Y is the query variable
  - Use probabilistic inference to compute most likely Y

\[
y = \arg \max_y P[y|f_1 \ldots f_n]
\]

- You already know how to do this inference

Simple Classification

- Simple example: two binary features

\[
P(m|s, f) = \frac{P(s, f|m)P(m)}{P(s, f)}
\]

Bayes estimate (no assumptions)

\[
P(m|s, f) = \frac{P(s|m)P(f|m)P(m)}{P(s, f)}
\]

Conditional independence

\[
\begin{align*}
P(+m, s, f) &= P(s + m)P(f + m)P(+m) \\
P(-m, s, f) &= P(s - m)P(f - m)P(-m)
\end{align*}
\]
General Naïve Bayes

- A general naïve Bayes model:
  \[ P(Y, F_1 \ldots F_n) = \prod_{i} P(F_i|Y) \]
  \[ |Y| \times |F| \times |Y| \text{ parameters} \]
- We only specify how each feature depends on the class
- Total number of parameters is linear in \( n \)

Inference for Naïve Bayes

- Goal: compute posterior over causes
- Step 1: get joint probability of causes and evidence
  \[ P(Y, f_1 \ldots f_n) = \frac{P(y_1, f_1 \ldots f_n) \ldots P(y_n, f_1 \ldots f_n)}{P(f_1 \ldots f_n)} \]
- Step 2: get probability of evidence
- Step 3: renormalize

General Naïve Bayes

- What do we need in order to use naïve Bayes?
  - Inference (you know this part)
    - Start with a bunch of conditionals, \( P(Y) \) and the \( P(F_i|Y) \) tables
    - Use standard inference to compute \( P(Y|F_1 \ldots F_n) \)
    - Nothing new here
  - Estimates of local conditional probability tables
    - \( P(Y) \), the prior over labels
    - \( P(F_i|Y) \) for each feature (evidence variable)
    - These probabilities are collectively called the parameters of the model and denoted by \( \theta \)
    - Up until now, we assumed these appeared by magic, but...
      - ...they typically come from training data: we’ll look at this now

A Digit Recognizer

- Input: pixel grids
- Output: a digit 0-9

Naïve Bayes for Digits

- Simple version:
  - One feature \( F_i \) for each grid position \(<i,j>\)
  - Possible feature values are on / off, based on whether intensity is more or less than 0.5 in underlying image
  - Each input maps to a feature vector, e.g.
    \[ \left\{ f_{3,0} = 0, f_{0,1} = 0, f_{0,2} = 1, f_{3,3} = 1, f_{1,4} = 0, \ldots f_{15,15} = 0 \right\} \]
  - Here: lots of features, each is binary valued
- Naïve Bayes model:
  \[ P(Y|F_{0,0} \ldots F_{15,15}) \propto P(Y) \prod P(F_{i,j}|Y) \]
- What do we need to learn?

Examples: CPTs
Parameter Estimation

- Estimating distribution of random variables like $X$ or $X | Y$
- Empirically: use training data
  - For each outcome $x$, look at the empirical rate of that value:
  $$p_{ML}(x) = \frac{\text{count}(x)}{\text{total samples}}$$
  $$p_{ML}(r) = 1/2$$
  - This is the estimate that maximizes the likelihood of the data
  $$L(x, \theta) = \prod_i p(x_i)$$

Elicitation: ask a human!

- Usually need domain experts, and sophisticated ways of eliciting probabilities (e.g. betting games)
- Trouble calibrating

Naïve Bayes for Text

- Bag-of-Words Naïve Bayes:
  - Predict unknown class label (spam vs. ham)
  - Assume evidence features (e.g. the words) are independent
  - Warning: subtly different assumptions than before!

Generative model
$$P(Y, W_1, \ldots, W_n) = P(Y) \prod_i P(W_i | Y)$$

Tied distributions and bag-of-words
- Usually, each variable gets its own conditional probability distribution $P(F|Y)$
- In a bag-of-words model
  - Each position is identically distributed
  - All positions share the same conditional probs $P(W|C)$
- Why make this assumption?

Spam Example

| Word | $P(w | \text{spam})$ | $P(w | \text{ham})$ | Tot Spam | Tot Ham |
|------|----------------------|----------------------|----------|---------|
| (prior) | 0.33333 | 0.66666 | -1.1 | -0.4 |

$P(\text{spam} | w) = 98.9$

A Spam Filter

- Naïve Bayes spam filter

Data:
- Collection of emails, labeled spam or ham
- Note: someone has to hand label all this data!
- Split into training, held-out, test sets

Classifiers
- Learn on the training set
- (Tune it on a held-out set)
- Test it on new emails

Example: Overfitting

$P(\text{spam}, Y = 2)$

<table>
<thead>
<tr>
<th>$P(\text{features}, Y = 2)$</th>
<th>$P(\text{features}, Y = 3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(Y = 2) = 0.1$</td>
<td>$P(Y = 3) = 0.1$</td>
</tr>
<tr>
<td>$P(\text{on}</td>
<td>Y = 2) = 0.8$</td>
</tr>
<tr>
<td>$P(\text{on}</td>
<td>Y = 2) = 0.1$</td>
</tr>
<tr>
<td>$P(\text{off}</td>
<td>Y = 2) = 0.1$</td>
</tr>
<tr>
<td>$P(\text{on}</td>
<td>Y = 2) = 0.01$</td>
</tr>
</tbody>
</table>

2 wins!!
Example: Overfitting

- Posterior determined by relative probabilities (odds ratios):

\[
\begin{align*}
P(W|\text{ham}) & \quad P(W|\text{spam}) \\
\text{south-west} & : \text{inf} & \text{screens} & : \text{inf} \\
\text{nation} & : \text{inf} & \text{minute} & : \text{inf} \\
\text{morally} & : \text{inf} & \text{guaranteed} & : \text{inf} \\
\text{nicely} & : \text{inf} & \$205.00 & : \text{inf} \\
\text{extent} & : \text{inf} & \text{delivery} & : \text{inf} \\
\text{seriously} & : \text{inf} & \text{signature} & : \text{inf} \\
\ldots & & \ldots & 
\end{align*}
\]

What went wrong here?

Generalization and Overfitting

- Relative frequency parameters will overfit the training data!
  - Just because we never saw a 3 with pixel (15,15) on during training doesn’t mean we won’t see it at test time.
  - Unlikely that every occurrence of “minute” is 100% spam.
  - Unlikely that every occurrence of “seriously” is 100% ham.
  - What about all the words that don’t occur in the training set at all?
  - In general, we can’t go around giving unseen events zero probability.

- As an extreme case, imagine using the entire email as the only feature.
  - Would get the training data perfect (if deterministic labeling).
  - Wouldn’t generalize at all.
  - Just making the bag-of-words assumption gives us some generalization, but isn’t enough.

- To generalize better: we need to smooth or regularize the estimates.

Estimation: Smoothing

- Problems with maximum likelihood estimates:
  - If I flip a coin once, and it’s heads, what’s the estimate for $P(\text{heads})$?
  - What if I flip 10 times with 8 heads?
  - What if I flip 10M times with 8M heads?

- Basic idea:
  - We have some prior expectation about parameters (here, the probability of heads).
  - Given little evidence, we should skew towards our prior.
  - Given a lot of evidence, we should listen to the data.

Estimation: Laplace Smoothing

- Laplace’s estimate:
  - Pretend you saw every outcome once more than you actually did.
  - Can derive this as a MAP estimate with Dirichlet priors (see cs281a).

- Laplace’s estimate (extended):
  - Pretend you saw every outcome $k$ extra times.
  - What’s Laplace with $k = 0$?
  - $k$ is the strength of the prior.

- Laplace for conditionals:
  - Smooth each condition if
  \[
P_{\text{LAP}}(x|y) = \frac{c(x,y) + k}{c(y) + k|X|}
\]
Estimation: Linear Interpolation

- In practice, Laplace often performs poorly for $P(X|Y)$:
  - When $|X|$ is very large
  - When $|Y|$ is very large

- Another option: linear interpolation
  - Also get $P(X)$ from the data
  - Make sure the estimate of $P(X|Y)$ isn’t too different from $P(X)$

$$P_{LIN}(x|y) = \alpha P(x|y) + (1.0 - \alpha) P(x)$$

- What if $\alpha$ is 0? 1?

- For even better ways to estimate parameters, as well as details of the math see cs281a, cs288

Real NB: Smoothing

- For real classification problems, smoothing is critical
- New odds ratios:

|          | $P(W|\text{ham})$ | $P(W|\text{spam})$ |
|----------|-------------------|-------------------|
| helvetica | 11.4              | 12.8              |
| sans      | 10.8              | 13.4              |
| group     | 10.2              | 14.2              |
| <FONT>   | 28.9              | 26.9              |

Do these make more sense?

Tuning on Held-Out Data

- Now we’ve got two kinds of unknowns
  - Parameters: the probabilities $P(Y|X), P(Y)$
  - Hyperparameters, like the amount of smoothing to do $k, \alpha$

- Where to learn?
  - Learn parameters from training data
  - Must tune hyperparameters on different data
  - Why?
    - For each value of the hyperparameters, train and test on the held-out data
    - Choose the best value and do a final test on the test data

Baselines

- First step: get a baseline
  - Baselines are very simple “straw man” procedures
  - Help determine how hard the task is
  - Help know what a “good” accuracy is

- Weak baseline: most frequent label classifier
  - Gives all test instances whatever label was most common in the training set
  - E.g. for spam filtering, might label everything as ham
  - Accuracy might be very high if the problem is skewed
  - E.g. calling everything “ham” gets 66%, so a classifier that gets 76% isn’t very good...

  - For real research, usually use previous work as a (strong) baseline

Confidences from a Classifier

- The confidence of a probabilistic classifier:
  - Posterior over the top label
  - $\text{confidence}(x) = \max_y P(y|x)$
  - Represents how sure the classifier is of the classification
  - Any probabilistic model will have confidences
  - No guarantee confidence is correct

- Calibration
  - Weak calibration: higher confidences mean higher accuracy
  - Strong calibration: confidence predicts accuracy rate
  - What’s the value of calibration?

Precision vs. Recall

- Let’s say we want to classify web pages as homepages or not
  - In a test set of 1K pages, there are 3 homepages
  - Our classifier says they are all non-homepages
  - 99.7 accuracy!
  - Need new measures for rare positive events

- Precision: fraction of guessed positives which were actually positive
- Recall: fraction of actual positives which were guessed as positive

- Say we guess 5 homepages, of which 2 were actually homepages
  - Precision: 2 correct / 5 guessed = 0.4
  - Recall: 2 correct / 3 true = 0.67

- Which is more important in customer support email automation?
- Which is more important in airport face recognition?
**Precision vs. Recall**

- Precision/recall tradeoff
  - Often, you can trade off precision and recall
  - Only works well with weakly calibrated classifiers
- To summarize the tradeoff:
  - **Break-even point**: precision value when \( p = r \)
  - **F-measure**: harmonic mean of \( p \) and \( r \):
    \[
    F_1 = \frac{2}{\frac{1}{p} + \frac{1}{r}}
    \]

**Errors, and What to Do**

- **Examples of errors**
  
  Dear GlobalSCAPE Customer,

  GlobalSCAPE has partnered with ScanSoft to offer you the latest version of OmniPage Pro, for just $99.99* - the regular list price is $499! The most common question we’ve received about this offer is - is it genuine? We would like to assure you that this offer is authorized by ScanSoft, is genuine and valid. You can get the . . .

  To receive your $30 Amazon.com promotional certificate, click through to http://www.amazon.com/apparel and see the prominent link for the $30 offer. All details are there. We hope you enjoyed receiving this message. However, if you’d rather not receive future e-mails announcing new store launches, please click . . .

**What to Do About Errors?**

- Need more features– words aren’t enough!
  - Have you emailed the sender before?
  - Have 1K other people just gotten the same email?
  - Is the sending information consistent?
  - Is the email in ALL CAPS?
  - Do inline URLs point where they say they point?
  - Does the email address you by (your) name?
- Can add these information sources as new variables in the NB model
- Next class we’ll talk about classifiers which let you easily add arbitrary features more easily

**Summary Naïve Bayes Classifier**

- **Bayes rule** lets us do diagnostic queries with causal probabilities
- The naïve Bayes assumption takes all features to be independent given the class label
- We can build classifiers out of a naïve Bayes model using training data
- Smoothing estimates is important in real systems
- Classifier confidences are useful, when you can get them